Pulling multiple nodes for rumor spreading

Frdrique Robin[1], Bruno Sericola[2], Emmanuelle Anceaume[3], and Yves Mocquard[4]

[1] Inria, Univ. Rennes, CNRS, IRISA
Inria, Campus de Beaulieu, 35042 Rennes Cedex, France
frederique.robin@inria.fr
[2] Inria, Univ. Rennes, CNRS, IRISA
Inria, Campus de Beaulieu, 35042 Rennes Cedex, France
bruno.sericola@inria.fr
[3] CNRS, Univ. Rennes, Inria, IRISA
IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France
emmanuelle.anceaume@irisa.fr
[4] Inria, Univ. Rennes, CNRS, IRISA
Inria, Campus de Beaulieu, 35042 Rennes Cedex, France
yves.mocquard@irisa.fr

Abstract. In this paper, we propose and analyze a new asynchronous rumor spreading protocol to deliver a rumor to all the nodes of a large-scale distributed network. This spreading protocol relies on what we call a k-pull operation, with $k \geq 2$. Specifically during a k-pull operation, an uninformed node i contacts $k-1$ random nodes in the network, and if at least one of them knows the rumor, then node i learns it. We perform a thorough study of $T_{k,n}$, the total number of k-pull operations needed for all the nodes to learn the rumor. We prove that the mean number of interactions needed for all the nodes to be informed is in $\mathcal{O}\left(n\ln(n)/(k-1)\right)$, which generalizes the standard case $k=2$ for the push-pull, push and pull protocols. We also analyze the tail of $T_{k,n}$ and prove that $T_{k,n} < c\mathbb{E}(T_{k,n})$ almost surely for any $c \in (0,1)$ when n tends to infinity. Finally, we prove that when $k > 2$, our new protocol requires less operations than the traditional push-pull or push (resp. pull) protocols by using stochastic dominance arguments.

Keywords: Rumor spreading · pull protocol · Markov chain · Asymptotic analysis

1 Introduction

This paper focuses on the dissemination of information from users to users in a decentralized manner. Peer-to-peer (P2P) networks allow users or more generally nodes to exchange information by relying on gossip protocols, also called randomized rumor spreading protocols. A randomized spreading rumor protocol describes the rules required for one or more pieces of information known to an arbitrary node in the network to be spread to all the nodes of the network. The push and pull protocols are the basic operations nodes use to propagate an information over the entire network [6, 9]. With the push operation, an informed node contacts some randomly chosen node and sends it the rumor, while with the pull operation, an uninformed node contacts some random node and asks for the rumor. The same node can perform both operations according to whether it knows or not the rumor, which corresponds to the push-pull protocol, or performs only one, either a pull or push operation, which corresponds to the pull or push protocols respectively. One of the important questions raised by these protocols is the spreading time, that is the time it needs for the rumor to be known by all the nodes of the network.

Several models have been considered to answer this question. The most studied one is the synchronous model. This model assumes that all the nodes of the network act in synchrony, which allows the algorithms designed in this model to divide time in synchronized rounds. During each synchronized round, each node i of the network selects at random one of its neighbor j and either sends to j the rumor if i knows it (push operation) or gets the rumor from j if j knows the rumor (pull operation). In the synchronous model, the spreading time of a rumor is defined as the number of synchronous rounds necessary for all the nodes to know the rumor. When the underlying graph is complete, it has been shown by Frieze [13] that the ratio of the number of rounds over $\log_2(n)$ converges in probability to $1 + \ln(2)$ when the number n of nodes in the graph tends to infinity. Further results have been established (see for example [19, 24] and the references therein), the most recent ones resulting from the observation that the rumor spreading time is closely related to the conductance of the graph of the network, see [15]. Investigations have also been done in different topologies of the network as in [2, 5, 12, 22], in the presence of link or nodes failures as in [11], in dynamic graphs as in [3] and spreading with node expansion as in [16]. Another alternative consists for the nodes to make more than one call during the push or pull operations [23]. This alternative is of particular interest since it does not require any particular network structure. The synchronous case has been tackled in [23] where the authors show that the push-pull protocol takes $\mathcal{O}(\log n / \log \log n)$ rounds in expectation assuming that nodes can connect to a random number neighbor following a specific power law during each single round.

In large scale distributed systems, assuming that all nodes act synchronously is a very strong assumption. Several authors, including [1, 8, 17, 20, 25]) suppose that nodes asynchronously trigger operations with randomly chosen nodes. Note that in [25], the authors model a multiple call by tuning the clock rate of each node with a given probability distribution. Moreover, some authors have focused on the message complexity by optimizing the network structure [7, 9, 17, 21]. For instance, in [7], the authors show that the asynchronous push-pull protocol spreading time in a preferential attachment graph is in $\mathcal{O}\left(\sqrt{\log n}\right)$. Another way of limiting the number of interactions is by finely tuning the push and pull operations to take advantage of both of them as achieved for example in [6, 10], or by relying on a central authority to coordinate the work (e.g, [4]).

The pull algorithm attracted very little attention because this operation was long considered inefficient to spread a rumor within a network [27]. It is actually very useful in systems fighting against message saturation (see for instance [29]). The ineffectiveness of the pull protocol stems from the fact that it takes some time before the rumour reaches a phase of exponential growth. Conversely, the push protocol initiates the rumor very quickly but then struggles to reach the few uninformed nodes. In this paper, we sought to counterbalance the slow initiation of rumour spreading by increasing the chances of learning the rumour with each call. Hence, in Section 2, we propose an asynchronous pull protocol during which the initiator of the operation calls the rumor from multiple nodes in parallel, and perform a thorough study $T_{k,n}$ the total number of pull operations needed for all the nodes to learn the rumor. We prove in Section 3 that the mean number of interactions needed to inform all the nodes of the system, assuming that one node knows initially the rumor, is in $\mathcal{O}\left(n\ln(n)/(k-1)\right)$. We then analyze the tail distribution of $T_{k,n}$ and prove that $T_{k,n} < c\mathbb{E}\left(T_{k,n}\right)$ almost surely for any $c \in (0,1)$ when n tends to infinity. In Section 4, we prove that when $k \geq 3$, our new protocol requires less interactions than the push-pull protocol [20] or the standard push (resp. pull) protocol by using stochastic dominance argument. Moreover, this efficiency increases strictly with k. Finally, we show that depending on the number of nodes that initially knows the rumor, the pull protocol always performs better than the push-pull one.

2 The k-pull protocol

2.1 Algorithm

We consider a complete network of size n in which each node may be asked for a piece of information (pull event). The algorithm starts with a single node informed of the rumor. At each time t, a uninformed node i contacts $k-1$ distinct nodes, chosen at random uniformly among the $n-1$ other nodes. If at least one of these contacted nodes knows the rumor, node i learns it. Pseudo-code of the algorithm is given in Algorithm 1. In the sequel, we analyze the distribution of the number of pull operations needed such that all the nodes are informed, and compare it to the standard (i.e. $k=2$) asynchronous push, pull and push-pull protocols.

Algorithm 1 pull operation run by any uninformed node i

$v(i) = 0$
upon event PULL ▷ PULL is triggered only if $v(i) = 0$
    select randomly $k-1$ nodes $i_1 \dots, i_{k-1}$
    if $\exists\, v(i_j) = 1$ **then**
        $v(i) \leftarrow 1$
    end if
end upon event

2.2 Discrete time Markov chain model

To analyze our k-pull algorithm, we introduce the stochastic process $Y := (Y_t)_{t\geq 0}$ where Y_t represents the number of informed nodes at discrete instant t. Stochastic process Y is a homogeneous Markov chain with n states where states $1, \ldots, n-1$ are transient and state n is absorbing.
From the algorithm, we deduce that when the Markov chain Y is in state i at time t, then at time $t+1$, either it remains in state i if none of the chosen nodes were informed or it transits to state $i+1$ otherwise. We denote by P its transition probability matrix. The non zero entries of matrix P are thus given for any $i = 1, \ldots, n-1$, by $P_{i,i}$ and $P_{i,i+1}$. Probability $P_{i,i+1}$ is given by

$$P_{i,i+1} = \frac{\displaystyle\sum_{j=\max\{1,k-n+i\}}^{\min\{i,k-1\}} \binom{i}{j}\binom{n-1-i}{k-1-j}}{\displaystyle\binom{n-1}{k-1}}.$$

This is the probability that a random variable, with the hypergeometric distribution with parameters $i, k-1, n-1$, is greater than or equal to 1. It follows easily using the Vandermonde equality that, for any $i = 1, \ldots, n-1$,

$$P_{i,i+1} = \begin{cases} 1 - \dfrac{\dbinom{n-1-i}{k-1}}{\dbinom{n-1}{k-1}} & \text{if } i \leq n-k \\ 1 & \text{otherwise.} \end{cases}$$

Obviously, we get, for any $i = 1, \ldots, n-1$,

$$P_{i,i} = \begin{cases} \dfrac{\dbinom{n-1-i}{k-1}}{\dbinom{n-1}{k-1}} & \text{if } i \leq n-k \\ 0 & \text{otherwise.} \end{cases}$$

We denote by $T_{k,n}$ the random variable defined by

$$T_{k,n} := \inf\{t \geq 0 \mid Y_t = n\}$$

which represents the spreading time, i.e. the total number of pull operations needed for all the nodes in the network to know the rumor.

As illustrated in Figure 1, when the Markov chain Y reaches state i, for $i = 1, \ldots, n-k$, it either transits to state $i+1$ with probability $P_{i,i+1}$ or remains in state i with probability $P_{i,i}$. In contrast, when Y reaches state $i = n-k+1$,

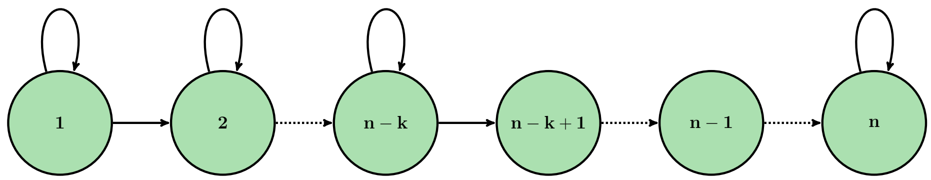

Fig. 1. Transition graph of Markov chain Y

an uninformed node almost surely selects at least one informed node among the $k-1$ nodes drawn from a population composed of $n-k+1$ informed ones and $k-2$ uninformed nodes ($P_{n-k+1,n-k+1} = 0$). This remains true for $i = n-k+1, \ldots, n-1$. The spreading time distribution $T_{k,n}$ can thus be expressed as a sum of independent random variables $S_{k,n}(i)$, where $S_{k,n}(i)$ is the sojourn time of process Y in state i. For all $i = 1, \ldots, n-k$, $S_{k,n}(i)$ follows a geometric distribution with parameter $p_{k,n}(i)$, where

$$p_{k,n}(i) := 1 - P_{i,i} = 1 - \prod_{h=1}^{k-1} \left(1 - \frac{i}{n-h}\right), \tag{1}$$

and $S_{k,n}(i) = 1$, for $i = n-k+1, \ldots, n-1$. Thus $T_{k,n}$ verifies

$$T_{k,n} = \sum_{i=1}^{n-1} S_{k,n}(i) = k - 1 + \sum_{i=1}^{n-k} S_{k,n}(i). \tag{2}$$

In the sequel, we analyze the two first moments of the discrete spreading time $T_{k,n}$ when n goes to infinity (Section 3), and compare our results with the standard push-pull, pull (case $k = 2$) and push protocols (Section 4). These analyses rely on a deep analysis of the sojourn times $S_{k,n}(i)$, $i = 1, \ldots, n-k$.

3 Moments of the asymptotic spreading time

In this section, we analyze the two first moments of the spreading time $T_{k,n}$ when n goes to infinity. For this purpose, we first look for bounds as tight as possible for the probabilities $p_{k,n}(i)$, $i = 1, \ldots, n-k$.

We introduce the function $P_{k,n}(x)$ defined for all $x \in \mathbb{R}$, the set of real numbers, for every $n \geq 3$ and $k = 1, \ldots, n-1$, by

$$P_{k,n}(x) := 1 - \prod_{h=1}^{k} \left(1 - \frac{x}{n-h}\right). \tag{3}$$

Note that

$$p_{k,n}(i) = P_{k-1,n}(i). \tag{4}$$

Lemma 1 *For all $x \in \mathbb{R}$, we have*

$$\frac{d}{dx}P_{k,n}(x) = \frac{(-1)^{k+1}k}{\prod_{h=1}^{k}(n-h)}\prod_{h=1}^{k-1}(x-\mu_h),$$

where, for all $h = 1, \ldots, k-1$, μ_h are positive real numbers such that $n-(h+1) < \mu_h < n-h$.

Proof. Note that for all $h = 1, \ldots, k$, we have $P_{k,n}(n-h) = 1$. Since $P_{k,n}(x)$ is a continuous function, it follows that there exists necessarily at least one local extremum point, denoted by μ_h, in each interval $(n-h-1, n-h)$, for $h = 1, \ldots, k-1$. The point μ_h is therefore a root of the polynomial $dP_{k,n}(x)/dx$. Note also that since the polynomial $1 - P_{k,n}(x)$ has only simple roots, we necessarily have $\mu_h \neq n-h-1$ and $\mu_h \neq n-h$. Using the fact that $P_{k,n}(x)$ is a k-degree polynomial, we deduce that $dP_{k,n}(x)/dx$ is a $(k-1)$-degree polynomial. The number of extremum μ_h being at least equal to $k-1$, this implies that the μ_h are unique. We thus first conclude that

$$\frac{d}{dx}P_{k,n}(x) = K\prod_{h=1}^{k-1}(x-\mu_h),$$

where K is a constant. We then conclude using the fact that the factor of the term x^k of polynomial $P_{k,n}(x)$ is equal to $(-1)^{k+1}/\prod_{h=1}^{k}(n-h)$. ■

Lemma 2 *For all $x \in [1, n-k]$, we have*

$$P_{k,n}(x) \leq \frac{kx}{n-k}.$$

Proof. From Lemma 1 and using the fact that $P_{k,n}(0) = 0$, we deduce that, for all $x \geq 0$,

$$P_{k,n}(x) = \frac{(-1)^{k+1}k}{\prod_{h=1}^{k}(n-h)}\int_0^x \prod_{h=1}^{k-1}(s-\mu_h)ds = \frac{k}{\prod_{h=1}^{k}(n-h)}\int_0^x \prod_{h=1}^{k-1}(\mu_h-s)ds.$$

Since $n-k < \mu_{k-1} < \cdots < \mu_1$, we get for all $x \in [1, n-k]$,

$$P_{k,n}(x) \leq \frac{kx}{\prod_{h=1}^{k}(n-h)}\max_{s\in[0,x]}\prod_{h=1}^{k-1}(\mu_h-s) = \frac{kx}{\prod_{h=1}^{k}(n-h)}\prod_{h=1}^{k-1}\mu_h.$$

Since $\mu_h < n-h$, for all $h = 1, \ldots, k-1$, we conclude that, for all $x \in [1, n-k]$,

$$P_{k,n}(x) \leq \frac{kx}{n-k},$$

which completes the proof. ■

We now turn to the lower bound of polynomial $P_{k,n}(x)$.

Lemma 3 *For all $x \in [1, n-k]$, we have*

$$\frac{kx}{n+kx} \leq P_{k,n}(x).$$

Proof. We first prove by recurrence that, for all integers $k \geq 1$, for all $x \in [1, n-k]$,

$$(n+kx)\prod_{h=1}^{k}(n-h-x) \leq \prod_{h=0}^{k}(n-h). \tag{5}$$

Relation (5) is true for $k=1$, since for all $x \in [1, n-1]$, we have $(n+x)(n-1-x) = n(n-1) - x - x^2 \leq n(n-1)$. Suppose now that Relation (5) is true at rank k. At rank $k+1$, using (5), we get, for all $x \in [1, n-k-1]$,

$$\begin{aligned}
(n+(k+1)x)\prod_{h=1}^{k+1}(n-h-x) &= (n-k-1-x)(n+(k+1)x)\prod_{h=1}^{k}(n-h-x) \\
&\leq (n-k-1-x)\left[1+\frac{x}{n+kx}\right]\prod_{h=0}^{k}(n-h) \\
&= \left[n-k-1-(k+1)\frac{x^2+x}{n+kx}\right]\prod_{h=0}^{k}(n-h) \\
&\leq (n-k-1)\prod_{h=0}^{k}(n-h) = \prod_{h=0}^{k+1}(n-h),
\end{aligned}$$

which proves Relation (5). Using now this relation, we obtain

$$1 - P_{k,n}(x) = \prod_{h=1}^{k}\left(1-\frac{x}{n-h}\right) = \frac{n\prod_{h=1}^{k}(n-h-x)}{\prod_{h=0}^{k}(n-h)} \leq \frac{n}{n+kx}.$$

This implies that $(kx)/(n+kx) \leq P_{k,n}(x)$, which ends the proof. ■

The expected value $\mathbb{E}(T_{k,n})$ is then easily obtained by

$$\mathbb{E}(T_{k,n}) = k-1+\sum_{i=1}^{n-k}\frac{1}{1-P_{i,i}} = k-1+\sum_{i=1}^{n-k}\frac{1}{p_{k,n}(i)}. \tag{6}$$

Theorem 4 (Asymptotic mean spreading time).

$$\mathbb{E}(T_{k,n}) \underset{n\longrightarrow\infty}{\sim} \frac{n\ln(n)}{k-1}.$$

Proof. Combining Relations (4) and (6), we get

$$\mathbb{E}(T_{k,n}) = k - 1 + \sum_{i=1}^{n-k} \frac{1}{P_{k-1,n}(i)}.$$

Applying now Lemmas 2 and 3, we obtain

$$k - 1 + \frac{n-k+1}{k-1} \sum_{i=1}^{n-k} \frac{1}{i} \leq \mathbb{E}(T_{k,n}) \leq n - 1 + \frac{n}{k-1} \sum_{i=1}^{n-k} \frac{1}{i}. \tag{7}$$

The fact that, for every $k \geq 0$, we have

$$\sum_{i=1}^{n-k} \frac{1}{i} \underset{n \longrightarrow \infty}{\sim} \ln(n),$$

completes the proof. ■

Concerning the variance, which is given by

$$\begin{aligned} \mathbb{V}ar(T_{k,n}) &= \sum_{i=1}^{n-k} \mathbb{V}ar(S_{k,n}(i)) = \sum_{i=1}^{n-k} \frac{1 - p_{k,n}(i)}{\left(p_{k,n}(i)\right)^2} \\ &= \sum_{i=1}^{n-k} \frac{1}{\left(p_{k,n}(i)\right)^2} - \left(\mathbb{E}(T_{k,n}) - k + 1\right), \end{aligned} \tag{8}$$

we have the following equivalent.

Theorem 5 (Asymptotic spreading time variance).

$$\mathbb{V}ar(T_{k,n}) \underset{n \longrightarrow \infty}{\sim} \frac{n^2}{(k-1)^2} \frac{\pi^2}{6}.$$

Proof. Applying Lemma 3, we get, from Relation (8),

$$\begin{aligned} \mathbb{V}ar(T_{k,n}) &\leq \sum_{i=1}^{n-k} \frac{1}{\left(p_{k,n}(i)\right)^2} \leq \sum_{i=1}^{n-k} \frac{(n + (k-1)i)^2}{(k-1)^2 i^2} \\ &= \frac{n^2}{(k-1)^2} \sum_{i=1}^{n-k} \frac{1}{i^2} + \frac{2n}{k-1} \sum_{i=1}^{n-k} \frac{1}{i} + n - k \underset{n \longrightarrow \infty}{\sim} \frac{n^2}{(k-1)^2} \frac{\pi^2}{6} \end{aligned}$$

Using Lemma 2 and applying Theorem 4, we obtain

$$\mathbb{V}ar(T_{k,n}) \geq \frac{(n-k+1)^2}{(k-1)^2} \sum_{i=1}^{n-k} \frac{1}{i^2} - \mathbb{E}(T_{k,n}) + k - 1 \underset{n \longrightarrow \infty}{\sim} \frac{n^2}{(k-1)^2} \frac{\pi^2}{6},$$

which completes the proof. ■

3.1 Bounds of the distribution $T_{k,n}$ and asymptotic analysis

Following the approach used in [20], we apply the bounds for the tail probabilities of a sum of independent, but not necessarily identically distributed, geometric random variables provided in [18] to deduce tail bounds for $T_{k,n}$, both for all n (Theorem 6) and when n tends to infinity (Theorem 7). We denote by H_n the Harmonic series defined, for every $n \geq 1$, by $H_n := \sum_{i=1}^{n} 1/i$.

Theorem 6. *1. For any $c \geq 1$,*

$$\mathbb{P}\left\{T_{k,n} \geq c\mathbb{E}\left(T_{k,n}\right)\right\} \leq \exp\left(-\frac{(k-1)^2 + (n-k)H_{n-k}}{n-1}(c-1-\ln(c))\right),$$

2. For any $c \leq 1$

$$\mathbb{P}\left\{T_{k,n} > c\mathbb{E}\left(T_{k,n}\right)\right\} \geq 1-\exp\left(-\frac{(k-1)^2 + (n-k)H_{n-k}}{n-1}(c-1-\ln(c))\right).$$

Proof. It is easily checked from Relation (1) that, for every $i = 1, \ldots, n-k$, we have

$$p_{k,n}(i) \geq p_{k,n}(1) = 1 - \prod_{h=1}^{k-1}\left(1 - \frac{1}{n-h}\right) = \frac{k-1}{n-1}.$$

We can now apply Theorem 11 (see Appendix) and deduce that for any $c \geq 1$,

$$\mathbb{P}\left\{T_{k,n} \geq c\mathbb{E}\left(T_{k,n}\right)\right\} \leq \exp\left(-\frac{k-1}{n-1}\mathbb{E}\left(T_{k,n}\right)(c-1-\ln(c))\right).$$

Note that $c - 1 - \ln(c) \geq 0$ for any $c > 0$. Using Relation (7), that is $\mathbb{E}\left(T_{k,n}\right) \geq k - 1 + (n-k)H_{n-k}/(k-1)$, we obtain, for any $c \geq 1$,

$$\mathbb{P}\left\{T_{k,n} \geq c\mathbb{E}\left(T_{k,n}\right)\right\} \leq \exp\left(-\frac{(k-1)^2 + (n-k)H_{n-k}}{n-1}(c-1-\ln(c))\right),$$

which concludes the first part of the proof.
From Theorem 12 (see Appendix), we deduce that for any $c \leq 1$,

$$\mathbb{P}\left\{T_{k,n} \leq c\mathbb{E}\left(T_{k,n}\right)\right\} \leq \exp\left(-\frac{k-1}{n-1}\mathbb{E}\left(T_{k,n}\right)(c-1-\ln(c))\right).$$

Since $c - 1 - \ln(c) \geq 0$ for $c > 0$, the same lower bound of $\mathbb{E}\left(T_{k,n}\right)$ used for the case $c \geq 1$, yields the same result. ■

Corollary 7 *For every $k \geq 2$, we have*

$$\lim_{n \longrightarrow +\infty} \mathbb{P}\left\{T_{k,n} > c\mathbb{E}\left(T_{k,n}\right)\right\} = \begin{cases} 0, & \text{if } c > 1 \\ 1, & \text{if } c < 1. \end{cases}$$

Proof. First, note that $c - 1 - \ln(c) > 0$ for any $c \in (0,1) \cup (1,\infty)$ and that $\lim_{n \longrightarrow +\infty} H_{n-k} = +\infty$. Hence,

$$\lim_{n \longrightarrow \infty} \exp\left(-\frac{(k-1)^2 + (n-k)H_{n-k}}{n-1}(c - 1 - \ln(c))\right) = 0.$$

Applying Theorem 6 concludes the proof. ∎

Corollary 7 implies that when n is large enough, $T_{k,n}$ distribution is concentrated between $k-1$ and $c\mathbb{E}(T_{k,n}) \underset{n \longrightarrow \infty}{\sim} cn\ln(n)/(k-1)$ for any $c > 1$ (according to Theorem 6). In fact, one can observe that for any $c \neq 1$,

$$\exp\left(-\frac{(k-1)^2 + (n-k)H_{n-k}}{n-1}(c - 1 - \ln(c))\right) \underset{n \longrightarrow \infty}{\sim} 1/n^{c-1-\ln(c)}$$

which indicates that $T_{k,n}$ distribution becomes closer to its mean at a speed of $1/n^{c-1-\ln(c)}$.
When $c = 1$, Corollary 7 does not allow us to figure out neither the existence of $\lim_{n \to +\infty} \mathbb{P}\{T_{k,n} \geq c\mathbb{E}(T_{k,n})\}$ nor its value.

4 Comparison of the *k*-pull with the push, pull and push-pull protocols

In this section, we compare the spreading time of the k-pull protocol with the standard (i.e. $k = 2$) push-pull, pull and push protocols ones. We summarize in Table 1 some characteristics of the spreading time distribution of each protocol. Note that the spreading time distributions of push and pull protocols are the same. In addition, the mean discrete spreading times of all the standard protocols (push, pull and push-pull) are the same.

	$T_{k,n}$ distribution	$\mathbb{E}(T_{k,n})$
push-pull	$\sum_{i=1}^{n-1} \mathcal{G}eom\left(\frac{2i(n-i)}{n(n-1)}\right)$	$(n-1)H_{n-1} \underset{n \longrightarrow \infty}{\sim} n \ln n$
push	$\sum_{i=1}^{n-1} \mathcal{G}eom\left(\frac{n-i}{n-1}\right)$	$(n-1)H_{n-1} \underset{n \longrightarrow \infty}{\sim} n \ln n$
pull	$\sum_{i=1}^{n-1} \mathcal{G}eom\left(\frac{i}{n-1}\right)$	$(n-1)H_{n-1} \underset{n \longrightarrow \infty}{\sim} n \ln n$
k-pull	$k - 1 + \sum_{i=1}^{n-k} \mathcal{G}eom(p_{k,n}(i))$	$\underset{n \longrightarrow \infty}{\sim} n\ln(n)/(k-1)$

Table 1. Asynchronous push, pull and push-pull spreading time distributions. The push-pull spreading time distribution is fully characterized in [20]. We detail the push case in Appendix A.

In this section, we denote by T_n^{push}, $T_n^{push-pull}$ and $T_{k,n}^{pull}$ for $k \geq 2$ the spreading time associated with respectively the push, push-pull and k-pull protocols.

4.1 Stochastic dominance of the *k*-pull protocol

To compare the spreading time distribution of each protocol, we use stochastic dominance tools. We recall the following definition (see [14]):

Definition 8 (Stochastic dominance definition) *Let X and Y two independent random variables.*

a) X strictly stochastically dominates Y if $\forall x \in \mathbb{R}$ $\mathbb{P}\{X > x\} > \mathbb{P}\{Y > x\}$. We then write $X \overset{s.t.}{\succ} Y$.
b) X stochastically dominates Y if $\forall x \in \mathbb{R}$ $\mathbb{P}\{X > x\} \geq \mathbb{P}\{Y > x\}$. We then write $X \overset{s.t.}{\succeq} Y$.

Comparing the spreading time distributions of each protocol amounts in comparing geometric distributions together. We thus first start by the following proposition.

Proposition 9 (Stochastic dominance for geometric law) *Let G_1 and G_2 two independent geometric random variables with parameters g_1 and g_2 respectively. We suppose that $g_1 > g_2$. Hence, G_1 is strictly stochastically dominated by G_2.*

Proof. Since $g_1 > g_2$, we have $(1-g_1)^\ell < (1-g_2)^\ell$ for every integer $\ell \geq 0$. Hence,

$$\mathbb{P}\left\{G_1 > \ell\right\} = (1-g_1)^\ell < (1-g_2)^\ell = \mathbb{P}\left\{G_2 > \ell\right\},$$

which implies that G_1 is strictly stochastically dominated by G_2. ∎

Theorem 10. *For all $n \geq k-1$,*

$$T_{k,n}^{pull} \overset{s.t.}{\prec} T_{k-1,n}^{pull} \overset{s.t.}{\prec} \ldots \overset{s.t.}{\prec} T_{3,n}^{pull} \overset{s.t.}{\prec} T_{2,n}^{pull} \overset{\mathcal{L}}{=} T_n^{push}$$

and

$$T_{3,n}^{pull} \overset{s.t.}{\prec} T_n^{push-pull}.$$

Proof. We first show that $p_{k,n}(i) > p_{k-1,n}(i)$ for all i. Note that for all $i = 1, \ldots, n-k$

$$1 - p_{k,n}(i) = \prod_{h=1}^{k-1}\left(1 - \frac{i}{n-h}\right) < \prod_{h=1}^{k-2}\left(1 - \frac{i}{n-h}\right) = 1 - p_{k-1,n}(i)$$

which implies that for all $i = 1, \ldots, n-k$, $p_{k,n}(i) > p_{k-1,n}(i)$. If $i = n-k+1$, $p_{k-1,n}(n-k+1) < 1$ and $p_{k,n}(n-k+1) = 1$.

Applying Proposition 9, we deduce $\mathcal{G}eom(p_{k,n}(i)) \overset{s.t.}{\prec} \mathcal{G}eom(p_{k-1,n}(i))$ for all $i = 1, \ldots, n-k$. Summing from $i = 1$ to $n-(k-1)$, we have

$$\sum_{i=1}^{n-k+1} \mathcal{G}eom\left(p_{k,n}(i)\right) \overset{s.t.}{\prec} \sum_{i=1}^{n-k+1} \mathcal{G}eom\left(p_{k-1,n}(i)\right).$$

Adding $k-2$ and since $p_{k,n}(n-k+1)=1$, we conclude that

$$T_{k,n}^{pull} \overset{s.t.}{\prec} T_{k-1,n}^{pull} \overset{s.t.}{\prec} \dots \overset{s.t.}{\prec} T_{2,n}^{pull}.$$

Note that from Table 1, the random variables $T_{2,n}^{pull}$ and T_n^{push} have the same distribution.

We turn now to the second part of the proof and first compare $T_{3,n}^{pull}$ to $T_n^{push-pull}$. Note that, for all $i=1,\dots,n-1$,

$$p_{2,n}(i) - \frac{2i(n-i)}{n(n-1)} = \frac{i}{n-1}\frac{(2n-3-i)n-2(n-i)(n-2)}{n(n-2)} = \frac{i(n+i(n-4))}{n(n-1)(n-2)}.$$

It is obvious that $n+i(n-4) \geq 0$ for all $i=1,\dots,n-2$ when $n \geq 4$. If $n=3$, then $n+i(n-4) \leq 3+1(3-4)=2>0$. Hence, we deduce from Proposition 9 that for all $i=1,\dots,n-1$,

$$\mathcal{G}eom\,(p_{2,n}(i)) \overset{s.t.}{\prec} \mathcal{G}eom\left(\frac{2i(n-i)}{n(n-1)}\right).$$

Summing from $i=1,\dots,n-1$, we conclude that $T_{3,n}^{pull} \overset{s.t.}{\prec} T_n^{push-pull}$, which ends the proof. ■

Theorem 10 shows that our k-pull protocol requires significantly less operations than the other standard protocols.

Figure 2 illustrates the fact that $T_{2,n}^{pull}$, T_n^{push} and $T_n^{push-pull}$ cannot be ordered with stochastic dominance arguments. Figure 2 shows that there is a threshold value $t(n)$ such that for all $t<t(n)$, $\mathbb{P}\{T_{2,n}^{pull}>t\} > \mathbb{P}\{T_n^{push-pull}>t\}$ and for all $t>t(n)$, $\mathbb{P}\{T_{2,n}^{pull}>t\} < \mathbb{P}\{T_n^{push-pull}>t\}$.

4.2 Choosing between push and pull

We now show that depending on the number of nodes that initially knows the rumor, the pull protocol always performs better than the push-pull one. This is again achieved by studying a particular combination of the sojourn times. Let i be the initial number of nodes knowing the rumor.

- If $i=1,\dots,\lfloor n/2\rfloor-1$, then $p_{2,n}(i)=i/(n-1) < 2i(n-i)/(n(n-1))$. In addition, $p_{2,n}(i)=2i(n-i)/(n(n-1))$ when $i=\lfloor n/2\rfloor$. Following the approach used in the proof of Theorem 10, we have

$$\sum_{i=1}^{\lfloor n/2\rfloor} \mathcal{G}eom(p_{2,n}(i)) \overset{s.t.}{\succ} \sum_{i=1}^{\lfloor n/2\rfloor} \mathcal{G}eom(2\frac{i(n-i)}{n(n-1)}).$$

- If $i=\lfloor n/2\rfloor+1,\dots,n-1$, then $p_{2,n}(i) \geq 2i(n-i)/(n(n-1))$. We have

$$\sum_{i=\lfloor n/2\rfloor+1}^{n-1} \mathcal{G}eom(p_{2,n}(i)) \overset{s.t.}{\prec} \sum_{i=\lceil n/2\rceil}^{n-1} \mathcal{G}eom(2i(n-i)/(n(n-1))).$$

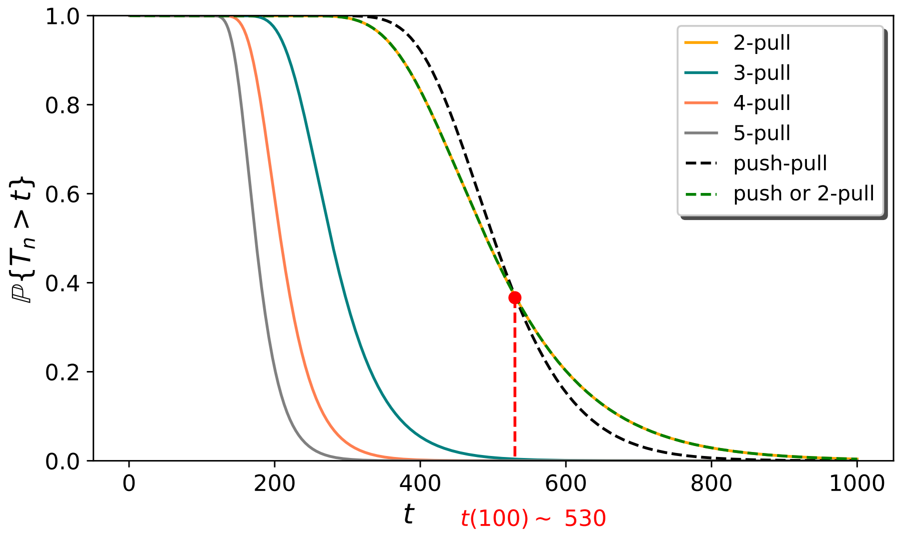

Fig. 2. Stochastic dominance illustration. Applying the algorithm detailed in Appendix B, we compute for the pull, push, push-pull and our k-pull protocols the tail probability $\mathbb{P}\{T_{k,n} > t\}$ for $n = 100$.

In other words, if more than a half the nodes are initially aware of the rumor, then the 2-pull protocol, and thus the k-pull protocol, will require significantly less operations than the push-pull protocol. The result is unclear when initially less than half of the nodes are aware of the rumor.

5 Conclusion

In this paper, we have proposed a new rumor spreading protocol that allows each node to asynchronously interact with $k-1$ other nodes during each operation. We have shown that this protocol generalizes the standard pull protocol and improves it when $k > 2$. Further research would allow us to manage competing rumours more finely. For instance, the initiator of the k-pull operation might take advantage of this interaction scheme to decide which rumor(s) to learn when different rumors compete. Dissemination of a rumor in a population pre-contaminated by two different rumors A and B has recently been studied by the Team-Rocket [26] in the context of blockchain protocols. In particular, assuming that rumor A (resp. B) is initially known by n_A nodes (resp. n_B nodes) with $n_A + n_B < n$, they leave as an open question the final proportion p_A of nodes knowing A (resp. p_B of nodes knowing B) with $p_A + p_B = 1$.

Appendix

Let $X_1, \ldots, X_n$ be n independent geometric random variables with possibly different distributions: $X_i \sim \mathcal{G}(p_i)$ with $p_i \in (0,1]$. Let $X = X_1 + \cdots + X_n$, $\mu = \mathbb{E}(X)$ and $p_* = \min_{i=1,\ldots,n} p_i$. Using these notations, the following result have been proved in Theorem 2.3 and Theorem 3.2 of [18] respectively.

Theorem 11. *For any $p_1, \ldots, p_n \in (0,1]$ and any $\lambda \geq 1$,*

$$\mathbb{P}\{X \geq \lambda\mu\} \leq e^{-p_*\mu(\lambda-1-\ln(\lambda))}.$$

Theorem 12. *For any $p_1, \ldots, p_n \in (0,1]$ and any $\lambda \leq 1$,*

$$\mathbb{P}\{X \leq \lambda\mu\} \leq e^{-p_*\mu(\lambda-1-\ln(\lambda))}.$$

A Push protocol

The continuous spreading time of the asynchronous push protocol has been computed in [14]. We quickly detail here the computation for the discrete spreading time.

We introduce a stochastic process $Z := (Z_t)_{t\geq 0}$ where Z_t corresponds to the number of informed nodes at discrete time t. Just as the process studied in [20] or our process (see Section 2.2), starting from state $i = 1, \ldots, n-1$, Markov chain Z either transits to state $i+1$ with probability $P^Z_{i,i+1} = (n-i)/(n-1)$ or stays in state i with probability $P^Z_{i,i} = 1 - (n-i)/(n-1)$. T_n^{push} can thus be expressed as the sum of the independent sojourn times $S^Z_n(i)$, where $S^Z_n(i)$ follows a geometric distribution of parameter $P^Z_{i,i+1}$. We deduce that

$$\mathbb{E}[T_n^{push}] = \sum_{i=1}^{n-1} \mathbb{E}[S^Z_n(i)] = \sum_{i=1}^{n-1} \frac{n-1}{n-i} = (n-1)H_{n-1}.$$

B Algorithm for the spreading time tail distribution

It is well-known, see for instance [28], that the distribution of $T_{k,n}$ is given, for every integer $t \geq 0$, by

$$\mathbb{P}\{T_{k,n} > t\} = \alpha Q^t \mathbb{1}, \tag{9}$$

where α is the row vector containing the initial probabilities of states $1, \ldots, n-1$, that is $\alpha_i = \mathbb{P}\{Y_0 = i\} = 1_{\{i=1\}}$, Q is the matrix obtained from the transition matrix P containing the transition probabilities between transient states and $\mathbb{1}$ is the column vector of dimension $n-1$ with all its entries equal to 1. Note that the submatrix Q of the transition probability matrix P, is upper triangular with a single non zero upper-diagonal, that is $Q_{i,j} = 0$ for all $i = 1, \ldots, n-1$ and $j \neq i, i+1$.

Following [20], we can propose an algorithm to compute the tail distribution of

$T_{k,n}$. Let $V(t) = (V_1(t), \ldots, V_{n-1}(t))$ be the column vector defined by $V_i(t) = \mathbb{P}\{T_{k,n} > t \mid Y_0 = i\}$. From Relation (9), we have $V(t) = Q^t \mathbb{1}$. Since $V(0) = \mathbb{1}$, writing $V(t) = QV(t-1)$ for $t \geq 1$, we get for any $t \geq 1$:

$$\begin{cases} V_i(t) = P_{i,i}V_i(t-1) + (1 - P_{i,i})V_{i+1}(t-1),\, i = 1, \ldots, n-2 \\ V_{n-1}(t) = P_{n-1,n-1}V_{n-1}(t-1). \end{cases}$$

The computation can be done using the following algorithm.

```
Input : integer ℓ
Output : P{T_{k,n} > 1}, ..., P{T_{k,n} > ℓ}
Initialization :
For i = 0 to ℓ do
    V_old[i] ← (P_{n-1,n-1})^i
EndFor
For i = n − 2 to 1 do
    V_new[0] ← 1 // corresponds to P{T_{k,n} > 0} = 1
    For t = 1 to ℓ do
        V_new[t] ← P_{i,i} V_new[t − 1] + (1 − P_{i,i}) V_old[t − 1]
    EndFor
    For t = 0 to ℓ do
        V_old[t] ← V_new[t]
    EndFor
EndFor
Return V
```

References

1. H. Acan, A. Collevecchio, A. Mehrabian, and N. Wormald. On the push & pull protocol for rumour spreading. *Trends in Mathematics*, 6:3–10, 2017.
2. F. Chierichetti, S. Lattanzi, and A. Panconesi. Rumor spreading in social networks. *Theoretical Computer Science*, 412(24):2602–2610, 2011.
3. A. Clementi, P. Crescenzi, C. Doerr, P. Fraigniaud, F. Pasquale, and R. Silvestri. Rumor spreading in random evolving graphs. *Random structures and Algorithms*, 48(2):290–312, 2015.
4. J. Correa, M. Kiwi, N. Olver, and A. Vera. Adaptive rumor spreading. *WINE 2015*, page 272285, 2016.
5. S. Daum, F. Kuhn, and Y. Maus. Rumor spreading with bounded indegree. In *Proceedings of the International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, 2016.
6. A. J. Demers, D. H. Greene, C. H. Hauser, W. Irish, J. Larson, S. J. Shenker, H. E. Sturgis, D. C. Swinehart, and D. B. Terry. Epidemic algorithms for replicated database maintenance. *PODC 87*, pages 1–12, 1987.
7. B. Doerr, M. Fouz, and T. Friedrich. Asynchronous rumor spreading in preferential attachment graphs. *SWAT 2012*, pages 307–315, 2012.
8. B. Doerr, M. Fouz, and T. Friedrich. Experimental analysis of rumor spreading in social networks. *MedAlg 2012*, pages 159–173, 2012.

9. B. Doerr and A. Kostrygin. Randomized Rumor Spreading Revisited. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *ICALP 2017*, volume 80, pages 138:1–138:14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017.
10. Z. Duan, K. Gopalan, and Y. Dong. Probabilistic reliable dissemination in large-scale systems. *SRUTI*, 2005.
11. F. Feige, D. Peleg, P. Raghavan, and E. Upfal. Randomized broadcast in networks. *Random Structures and Algorithms*, 1(4):447–460, 1990.
12. N. Fountoulakis and K. Panagiotou. Rumor spreading on random regular graphs and expanders. *Random Structures and Algorithms*, 43(2):201–220, 2013.
13. A. Frieze and G. Grimmet. The shortest-path problem for graphs with random arc-lengths. *Discrete Applied Mathematics*, 10(1):57–77, 1985.
14. A. J. Ganesh. Rumour spreading on graphs. *Technical report:https://people.maths.bris.ac.uk/ maajg/teaching/complexnets/rumours.pdf*, 2015.
15. G. Giakkoupis. Tight bounds for rumor spreading in graphs of a given conductance. In *Proceedings of the International Symposium on Theoretical Aspects of Computer Science (STACS)*, 2011.
16. G. Giakkoupis. Tight bounds for rumor spreading with vertex expansion. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2014.
17. G. Giakkoupis, Y. Nazari, and P. Woelfel. How asynchrony affects rumor spreading time. *PODC '16*, pages 185–194, 2016.
18. S. Janson. Tail bounds for sums of geometric and exponential variables. *Statistics & Probability Letters*, 135:1–6, 2018.
19. R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking. Randomized rumor spreading. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, 2000.
20. Y. Mocquard, S. Robert, B. Sericola, and E. Anceaume. Analysis of the propagation time of a rumour in large-scale distributed systems. *NCA 2016*, 2016.
21. Y. Moreno, M. Nekovee, and A. F. Pacheco. Dynamics of rumor spreading in complex networks. *Phys. Rev.*, 69:248–258, 2004.
22. K. Panagiotou, X. Perez-Gimenez, T. Sauerwald, and H. Sun. Randomized rumor spreading: the effect of the network topology. *Combinatorics, Probability and Computing*, 24(2):457–479, 2015.
23. K. Panagiotou, A. Pourmiri, and T. Sauerwald. Faster rumor spreading with multiple calls. *The electronic journal of combinatorics*, 22, 2015.
24. B. Pittel. On spreading a rumor. *SIAM Journal on Applied Mathematics*, 47(1):213–223, 1987.
25. A. Pourmiri and F. Ramezani. Brief announcement: Ultra-fast asynchronous randomized rumor spreading. *SPAA 2019*, 2019.
26. Team Rocket. Snowflake to avalanche: A novel metastable consensus protocol family for cryptocurrencies. *Technical report*, 2019.
27. S. Sanghavi, B. Hajek, and L. Massouli. Gossiping with multiple messages. *IEEE Transactions on Information Theory*, 53(123), 2007.
28. B. Sericola. *Markov Chains. Theory, Algorithms and Applications*. John Wiley & Sons, 2013.
29. G. Yao, J. Bi, S. Wang, Y. Zhang, and Y. Li. A pull model IPv6 duplicate address detection. *LCN 2010*, 2010.